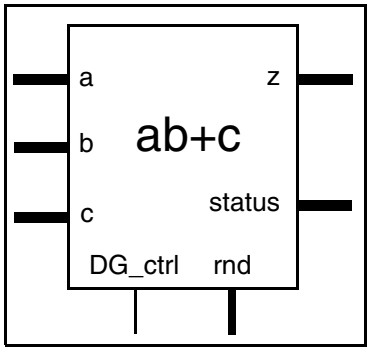
DW_fp_mac_DG

Floating-Point Multiply-and-Add with Datapath Gating

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History



- Has the same functionality as DW_fp_mac when the component is in normal operation
- Consumes less dynamic power than DW_fp_mac when disabled (inputs are active, but the output is not being used)
- The precision format is parameterizable for either IEEE single, double precision, or a user-defined custom format
- Hardware for denormal numbers of IEEE 754 standard is selectively provided
- Accuracy conforms to IEEE 754 Floating-point standard
- DesignWare datapath generators are employed for better power and QoR

Description

DW_fp_mac_DG is a floating-point component that performs the multiply-and-add operation. It sums up a floating-point product of input `a` and `b` to input `c`: `ab + c` to produce a floating-point multiply and add result, `z`. Also, a control input (`DG_ctrl`) can disable the component to reduce dynamic power consumption when the component is not in use and inputs are still active.

The output of this component has accuracy consistent with the IEEE Standard 754, where the computation happens as it was done using infinite precision, and rounding is executed as a last step to obtain the final result. As a consequence, the accuracy of DW_fp_mac_DG is much better than the accuracy of an implementation using DW_fp_mult(_DG) and DW_fp_add(_DG).

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	(*sig_width* + *exp_width* + 1) bits	Input	Multiplier
b	(*sig_width* + *exp_width* + 1) bits	Input	Multiplicand
c	(*sig_width* + *exp_width* + 1) bits	Input	Addend
rnd	3 bits	Input	Rounding mode; supports all rounding modes described in the *Datapath Floating-Point Overview*
z	(*sig_width* + *exp_width* + 1) bits	Output	Mac result (`a` x `b` + `c`)

Table 1-1 Pin Description (Continued)

Pin Name	Width	Direction	Function
status	8 bits	Output	Status flags for the result `z` For details, see STATUS Flags in the *Datapath Floating-Point Overview.*
DG_ctrl	1 bit	Input	Datapath gating control ■ 0: Component is disabled ■ 1: Normal component operation For details, see "Datapath Gating Control with DG_ctrl" on page 3.

Table 1-2 Parameter Description

Parameter	Values	Description
sig_width	2 to 253 bits	Word length of fraction field of floating-point numbers `a`, `b`, c, and `z`
exp_width	3 to 31 bits	Word length of biased exponent of floating-point numbers `a`, `b`, c and `z`
ieee_compliance	0 or 1	Level of support for IEEE 754: ■ 0: No support for IEEE 754 NaNs and denormals; NaNs are considered infinities and denormals are considered zeros ■ 1: Fully compliant with the IEEE 754 standard, including support for NaNs and denormals For more, see IEEE 754 Compatibility in the *Datapath Floating-Point Overview*.

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Datapath gating close to the main inputs	■ DesignWare (P-2019.03 and later) ■ DesignWare-LP (before P-2019.03)
str2[a]	Datapath gating to allow late arrival time of `DG_ctrl` signal	■ DesignWare (P-2019.03 and later) ■ DesignWare-LP (before P-2019.03)

a. By default, the rtl2 implementation is used for synthesis (see "Datapath Gating Control with DG_ctrl" on page 3).

Table 1-4 Simulation Models

Model	Function
DW02.DW_FP_MAC_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_fp_mac_sim.vhd	VHDL simulation model source code

Table 1-4 Simulation Models (Continued)

Model	Function
dw/sim_ver/DW_fp_mac.v	Verilog simulation model source code

Datapath Gating Control with DG_ctrl

For DW_fp_mac_DG and other combinational components that have the datapath gating feature, the `DG_ctrl` port is provided to control datapath gating.

When `DG_ctrl` = 1, the component behaves as expected according to activity on the input ports.

When `DG_ctrl` = 0:

- The component is disabled and internal gates are totally or partially isolated to block propagation of switching activity inside the component. This makes the component less sensitive to switching activity on the main ports and reduces dynamic power consumption.
- Values at the output ports are not defined.
- Simulation models set 'X' values at the output ports.

The implementations for DW_fp_mac_DG (see Table 1-3 on page 2) perform datapath gating differently:

- The str implementation places datapath gating as close as possible to the input ports to maximize dynamic power savings when `DG_ctrl` = 0. However, if the `DG_ctrl` signal arrives later than the data inputs, timing is degraded and area is increased to recover timing, which can increase power.
- The str2 implementation places datapath gating near the middle of the component. This approach is less sensitive to the arrival time of `DG_ctlr` and has a better chance of meeting timing and still providing dynamic power savings.

By default, the synthesis tool uses the str2 implementation, but you can override that. If timing constraints are loose or you know that the signal driving the `DG_ctrl` port arrives at the same time as other input signals, greater power savings can be attained by using the str implementation. You can make the override on a global level or on a case-by-case basis, as explained next.

To use the str implementation globally, you can disable the str2 implementation as follows:

- Design Compiler (before version P-2019.03):

```
set_dont_use {dw_minpower.sldb/DW_fp_div_DG/str2}
```

- Design Compiler (P-2019.03 and later)

```
set_dont_use {dw_foundation.sldb/DW_fp_div_DG/str2}
```

- Fusion Compiler:

```
set_synlib_dont_use {dw_foundation/DW_fp_div_DG/str2}
```

To use the str implementation for specific instantiated components, use the set_implementation command:

```
set_implementation U1 str
```

Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If an invalid rounding mode has been detected on `rnd`, the following message is displayed:

  ```
  WARNING: <instance_path>:
      at time = <timestamp>: Illegal rounding mode.
  ```

  To suppress this message, use the DW_SUPPRESS_WARN macro explained earlier.

Related Topics

- Datapath Floating-Point Overview
- DesignWare Building Blocks User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_Foundation_comp.all;

entity DW_fp_mac_DG_inst is
    generic (
          inst_sig_width : POSITIVE := 23;
          inst_exp_width : POSITIVE := 8;
          inst_ieee_compliance : INTEGER := 0
          );
    port (
          inst_a : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
          inst_b : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
          inst_c : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
          inst_rnd : in std_logic_vector(2 downto 0);
          inst_DG_ctrl : in std_logic;
          z_inst : out std_logic_vector(inst_sig_width+inst_exp_width downto 0);
          status_inst : out std_logic_vector(7 downto 0)
    );
end DW_fp_mac_DG_inst;

architecture inst of DW_fp_mac_DG_inst is

begin

  -- Instance of DW_fp_mac_DG
  U1 : DW_fp_mac_DG
       generic map ( sig_width => inst_sig_width, exp_width => inst_exp_width,
ieee_compliance => inst_ieee_compliance )
       port map ( a => inst_a, b => inst_b, c => inst_c, rnd => inst_rnd, DG_ctrl =>
inst_DG_ctrl, z => z_inst, status => status_inst );

end inst;

-- pragma translate_off
configuration DW_fp_mac_DG_inst_cfg_inst of DW_fp_mac_DG_inst is
for inst
end for; -- inst
end DW_fp_mac_DG_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_fp_mac_DG_inst( inst_a, inst_b, inst_c, inst_rnd, inst_DG_ctrl,
          z_inst, status_inst );

parameter sig_width = 23;
parameter exp_width = 8;
parameter ieee_compliance = 0;


input [sig_width+exp_width : 0] inst_a;
input [sig_width+exp_width : 0] inst_b;
input [sig_width+exp_width : 0] inst_c;
input [2 : 0] inst_rnd;
input inst_DG_ctrl;
output [sig_width+exp_width : 0] z_inst;
output [7 : 0] status_inst;

    // Instance of DW_fp_mac_DG
    DW_fp_mac_DG #(sig_width, exp_width, ieee_compliance)
      U1 ( .a(inst_a), .b(inst_b), .c(inst_c), .rnd(inst_rnd), .DG_ctrl(inst_DG_ctrl),
.z(z_inst), .status(status_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted the description of the *ieee_compliance* parameter in Table 1-2 on page 2 ■ Added "Suppressing Warning Messages During Verilog Simulation" on page 4
March 2019	DWBB_201903.0	■ Added "Datapath Gating Control with DG_ctrl" on page 3 ■ Clarified some information about minPower ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com